Procesamiento del Lenguaje Natural, núm. 32 (2004), pp. 1-8 recibido 6-12-2003; aceptado 20-02-2004

Análisis Comparativo de Lenguajes Basados en XML para el Desarrollo de Sistemas de Diálogo

R. López-Cózar

Dpto. Lenguajes y Sistemas Informáticos, E.T.S. Ingeniería Informática
18071 Universidad de Granada, Tel.: +34 958 240579, Fax: +34 958 243179
E-mail: rlopezc@ugr.es

Resumen: En este artículo se presenta un análisis comparativo de las principales ventajas e inconvenientes de los lenguajes VoiceXML, CCXML, CallXML y SALT, desarrollados en los últimos años para facilitar la implementación de sistemas de diálogo. El análisis se inicia con un estudio del lenguaje VoiceXML, considerado estándar y utilizado como referencia en el artículo para realizar las comparaciones. Para cada lenguaje se muestran ejemplos de aplicaciones sencillas con objeto de mostrar la sintaxis, incluyendo asimismo enlaces de Internet a sus respectivas especificaciones, donde los lectores pueden encontrar información detallada y completa de cada uno de ellos.
Palabras clave: VoiceXML, CCXML, CallXML, SALT, sistemas de diálogo, estándares.

Abstract: This paper presents a comparative analysis of the most relevant advantages and drawbacks of languages VoiceXML, CCXML, CallXML and SALT, recently developed to ease the setting up of dialogue systems. The analysis starts with an study of VoiceXML since it is considered a standard and it is used in the paper to compare the other languages with. The paper also presents simple applications coded in these languages to show the syntax as well as Internet links to their respective specifications, allowing readers can find detailed and complete information.
Keywords: VoiceXML, CCXML, CallXML, SALT, dialogue systems, standards.

1 Introducción

Los sistemas de diálogo son programas informáticos capaces de emular a un ser humano en un diálogo oral con otra persona. Actualmente, diversas empresas e instituciones públicas y privadas usan estos sistemas con la finalidad de proporcionar información y otros servicios de forma automática. Entre sus principales se pueden mencionar las siguientes: información y reserva de viajes en avión (Seneff y Polifroni, 2000), información meteorológica (Zue et al. 2000), compra de productos (López-Cózar et al. 1997), información y reserva de viajes en tren (Billi et al. 1997), etc.

En la década de los años ochenta los diseñadores de los sistemas de diálogo debían programar dichos sistemas manualmente, dada la ausencia de herramientas de desarrollo. Ello significaba tener que controlar a nivel de señal la voz proveniente de los usuarios, el flujo de la interacción, los posibles eventos que podían producirse, etc. En los años noventa aparecen diversas herramientas de desarrollo, como HTK (*Hidden Markov Model Toolkit*) (Hain et al., 1999) y CSLU Toolkit[1], que facilitan la implementación de tales sistemas, permitiendo que los desarrolladores se concentren en la lógica de las aplicaciones dejando al margen cuestiones de bajo nivel, con el consiguiente ahorro de esfuerzo y tiempo. También en la década de los años noventa aparecen en el mercado servidores web con capacidad de soportar la voz humana, propiciando la aparición de lenguajes como VoiceXML, CCXML, CallXML y SALT, basados en el lenguaje de etiquetas XML (*eXtensible Markup Language*). La finalidad de estos nuevos lenguajes es facilitar la implementación de los sistemas de diálogo aprovechando la infraestructura las ventajas ofrecidas por los sistemas de transmisión de información en Internet.

El artículo está estructurado de la siguiente forma. La sección 2 proporciona una visión general del lenguaje VoiceXML

[1] http://cslu.cse.ogi.edu/toolkit

ISSN 1135-5948

centrada en su arquitectura, ventajas y limitaciones. La sección 3 está dedicada al lenguaje CCXML, desarrollado para resolver algunas de las limitaciones de VoiceXML. El artículo menciona algunas de sus características y presenta sus principales diferencias respecto a VoiceXML. La secciones 4 y 5 se centran en los lenguajes CallXML y SALT, presentando igualmente sus principales características y diferencias respecto a VoiceXML. Finalmente, la sección 6 presenta las conclusiones del artículo.

2 VoiceXML

VoiceXML[2] (*Voice Extensible Markup Language*) es un lenguaje basado en etiquetas y similar a HTML, que ha sido desarrollado por el W3C (*World Wide Web Consortium*) para facilitar la implementación de sistemas de acceso telefónico oral y mediante DTMF (*Dial Tone Multi Frequency*), al contenido de las páginas web (Abbott, 2002). Este tipo de sistemas es muy importante para las empresas ya que permite reducir el número de llamadas que han de ser atendidas por operadores humanos, reduciendo por tanto los costes comerciales y los tiempos de espera de los clientes.

Los sistemas implementados mediante VoiceXML pueden usar voz sintetizada o ficheros de voz pregrabados para interactuar con los usuarios, mientras que éstos pueden usar su voz o pulsar los botones del teléfono para comunicarse con tales sistemas. A modo ilustrativo, la Figura 1 muestra un breve fragmento de código escrito en VoiceXML para solicitar el número de teléfono del usuario, reconocerlo mediante un proceso de reconocimiento automático del habla (RAH) (Rabiner y Juang, 1993), almacenarlo en una determinada variable para su posterior uso y generar un mensaje de ayuda a petición del usuario. Como se puede observar en el ejemplo, VoiceXML presenta la ventaja de aislar al implementador de las cuestiones de bajo nivel relacionadas con el RAH y la síntesis de habla (Campillo y Banga, 2003).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="1.0">
  <form id="datos_usuario">
    <field name="telefono" type="digits">
        <prompt>Diga su número de teléfono
        dígito a dígito </prompt>
        <help> Debe pronunciar el número
        dígito a dígito, por ejemplo, 9, 5,
        8, 1, 3, 1, 4, 2, 6 </help>
    </field>
  </form>
</vxml>
```

Figura 1. Ejemplo de código VoiceXML

Respecto al RAH, el implementador sólo ha de especificar qué tipo de datos se desea obtener del usuario y gestionar, a alto nivel, algunos eventos que se puedan producir durante dicho proceso (p. e. cuando el usuario solicita ayuda o no responde en un determinado intervalo de tiempo, etc.). Respecto a la síntesis de habla, el implementador sólo ha de especificar el mensaje a generar, en forma de texto o mediante un fichero de voz pregrabado.

A principios de 2001, VoiceXML fue considerado por el W3C el lenguaje estándar basado en etiquetas para crear este tipo de sistemas. Poco después, el VBWG (*Voice Browser Working Group*), un subgrupo de trabajo del W3C dedicado a la evolución del estándar, comenzó a trabajar en la versión 2.0, cuya versión inicial apareció en octubre de 2001.

2.1 Arquitectura de VoiceXML

El estándar VoiceXML se basa en la arquitectura modular mostrada en la Figura 2, constituida por los siguientes componentes: intérprete de VoiceXML, servidor de documentos, entorno del intérprete y plataforma de implementación. El intérprete de VoiceXML (aplicación cliente) procesa los comandos del documento VoiceXML, genera prompts, escucha las respuestas del usuario y ejecuta la lógica de la aplicación. Generalmente realiza consultas a diversos sitios web para obtener la información que se ha de proporcionar al usuario.

[2] http://www.voicexml.org

El servidor de documentos (generalmente un servidor web) procesa las peticiones enviadas por la aplicación cliente y proporciona como respuesta documentos VoiceXML.

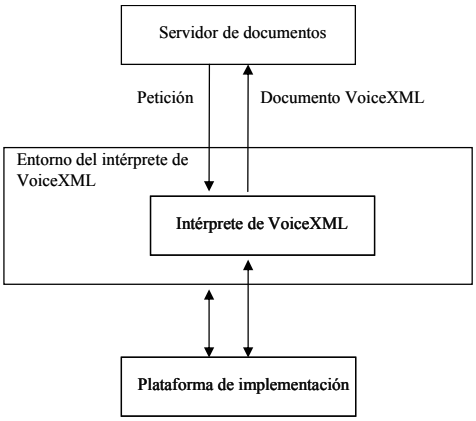

Figura 2. Arquitectura VoiceXML

El entorno del intérprete de VoiceXML procesa el documento y responde a las llamadas de los usuarios, monitorizando sus entradas en paralelo con el intérprete; por ejemplo, puede detectar frases específicas que el usuario puede utilizar para solicitar ayuda, cambiar las características de la conversión texto-habla, etc.

La plataforma de implementación contiene el hardware telefónico y otros recursos relacionados con el mismo, siendo su misión generar eventos en respuesta a las acciones del usuario (p. e. pulsaciones de botones u órdenes expresadas mediante voz).

2.2 Ventajas de VoiceXML

VoiceXML es una tecnología independiente de la plataforma de implementación y está basada en la misma arquitectura cliente/servidor utilizada por las aplicaciones HTML. De hecho, los sistemas basados en VoiceXML suelen ejecutarse conjuntamente con aplicaciones web tradicionales, accediendo a los mismos datos y realizando esencialmente las mismas tareas; incluso suelen estar ubicadas en las mismas máquinas. No obstante, al igual que no existe un lenguaje de programación óptimo para implementar cualquier tipo de programa, no existe una plataforma óptima para desarrollar cualquier tipo de aplicación basada en procesamiento del habla, existiendo una gran diversidad de factores que determinan qué arquitectura, hardware y sistema operativo se deben usar. VoiceXML no es una excepción; es una herramienta excelente para implementar aplicaciones telefónicas basadas en diálogos y entradas del usuario realizadas mediante voz o DTMF. No obstante, tales aplicaciones deben requerir una interacción reducida con la red telefónica, limitándose a responder a las llamadas de los usuarios, interactuar con ellos y desconectar la línea al finalizar la llamada. Si se necesita una gestión más sofisticada de la línea telefónica, para permitir por ejemplo la identificación de los números de teléfono o la tarifación de las llamadas, VoiceXML no es la opción más adecuada.

2.3 Limitaciones de VoiceXML

A pesar de sus numerosas ventajas, VoiceXML presenta también diversos inconvenientes que limitan su uso para un mayor rango de aplicaciones potenciales. Por ejemplo, la independencia de la plataforma de implementación conlleva que esté disponible un conjunto muy reducido de funciones telefónicas, no siendo posible controlar la velocidad o el volumen de los ficheros de audio utilizados, ni reproducirlos comenzando en un punto determinado (característica necesaria si se desea reanudar un mensaje de voz interrumpido mediante una pausa). En cambio, muchos sistemas de lectura de correo electrónico mediante voz permiten que los usuarios puedan pulsar una tecla para avanzar hacia delante o hacia atrás en el mensaje.

Por otra parte, VoiceXML es una tecnología basada exclusivamente en el habla (y en DTMF, en el caso de la entrada), siendo el teléfono el único dispositivo que pueden utilizar los usuarios. En cambio, usando otros lenguajes (p. e. SALT) se pueden implementar sistemas multimodales que permiten usar, adicionalmente, PDAs (*Personal Digital Assistant*), navegadores web y otros dispositivos de acceso a Internet. Entre estos sistemas se pueden mencionar, por ejemplo, los sistemas de navegación

instalados en automóviles que muestran en forma de texto o en un mapa las direcciones solicitadas por los conductores. Este tipo de sistemas no puede ser implementado mediante VoiceXML.

3 CCXML

VoiceXML no es tan robusto como se esperaba para satisfacer las necesidades comerciales. El lenguaje es muy apropiado para gestionar la interacción usuario-sistema mediante diálogos, pero carece de la funcionalidad necesaria para gestionar el control de las llamadas, cuestión ésta crítica para un gran número de sistemas comerciales. Por ejemplo, la versión 2.0 proporciona métodos para realizar la transferencia de llamadas, pero estos métodos son rudimentarios y tienen ciertas limitaciones; por ejemplo, una vez realizada una transferencia, la aplicación se da por terminada y el usuario no puede volver a interactuar con ella. Asimismo, la versión 2.0 tampoco permite poner en espera una llamada para realizar una presentación de la persona que realiza dicha llamada. Ello ha motivado que el VBWG proponga otro lenguaje basado en etiquetas, denominado CCXML[3] (*Call Control eXtensible Markup Language*), para aportar la funcionalidad de que carece VoiceXML en cuanto a control de llamadas, haciéndose pública la primera versión de este nuevo lenguaje en 2002.

CCXML ha sido diseñado para resolver las limitaciones de VoiceXML pero no para sustituirlo; de hecho, ambos lenguajes se complementan entre sí, proporcionando a los diseñadores las facilidades de desarrollo de diálogos proporcionadas por VoiceXML, junto con la funcionalidad del control de llamadas proporcionada por CCXML. De esta forma, los sistemas de diálogo pueden incorporar funciones telefónicas complejas, como las siguientes:

Encuéntreme – sígame. Permite localizar a una persona en varias ubicaciones posibles mediante un único número de teléfono, lo que evita que esta persona tenga que recordar varios números y las personas que la llamen tengan que efectuar varias llamadas.

Re-transferencia de llamadas. Permite que las llamadas puedan ser transferidas a otras personas varias veces.

Transferencia supervisada. Permite poner en espera las llamadas mientras se realiza una introducción de la persona que realiza la llamada.

Multiconferencia. Permite que varias personas puedan hablar simultáneamente entre sí.

A modo de ejemplo, la Figura 3 muestra una aplicación sencilla que acepta o rechaza una llamada telefónica en función del número de teléfono de destino, generando un mensaje de depuración u otro según el caso. Las etiquetas <eventhandler> ... </eventhandler> contienen el código para la gestión asíncrona de eventos. Las etiquetas <transition> se usan para capturar eventos y ejecutar acciones en respuesta a dichos eventos, por ejemplo, el evento connection.CONNECTION_ALERTING se genera cuando se detecta una llamada entrante, y el evento connection.CONNECTION_CONNECTED se genera cuando la llamada comienza a ser atendida. La etiqueta <log> se utiliza para generar mensajes de depuración, la etiqueta <reject/> se usa para rechazar la llamada, y la etiqueta <accept/> se usa para aceptarla.

```
<?xml version=”1.0” encoding=”UTF-8”?>
<ccxml version=”1.0”>
  <eventhandler>

    <transition
event=”connection.CONNECTION_ALERTING”
name=”evt”>
      <log expr=”"El ID del número llamado es ' +
evt.calledid + '."”””/>
      <if cond=”evt.calledid == '8315551234”’>
        <reject/>
       <else/>
        <accept/>
      </if>
    </transition>

    <transition
event=”connection.CONNECTION_CONNECTED”
>
      <log expr=”"La llamada está siendo atendida."”/>
      <disconnect/>
    </transition>

  </eventhandler>
</ccxml>
```

Figura 3. Ejemplo de código CCXML

[3] http://www.w3.org/TR/ccxml/

El funcionamiento de la aplicación es muy sencillo. Cuando se detecta la llamada entrante se genera un mensaje de depuración que informa del número de teléfono destino de la llamada. Seguidamente se comprueba este número, aceptándose la llamada si coincide con el número "8315551234" y rechazándose en caso contrario. Finalmente, si la llamada es aceptada se genera otro mensaje de depuración que informa que la llamada es atendida, y se realiza la desconexión de la llamada.

3.1 Diferencias respecto a VoiceXML

Mientras que VoiceXML ha sido diseñado principalmente para implementar diálogos entre usuarios y aplicaciones web, CCXML ha sido diseñado para implementar aplicaciones telefónicas complejas, con posibilidad de lanzar nuevas aplicaciones IVR (*Interactive Voice Reponse*) en llamadas subsiguientes, cuestión ésta que se puede realizar de forma muy rudimentaria mediante VoiceXML. CCXML ha sido diseñado con la finalidad de complementar a VoiceXML y mejorar así la funcionalidad de los diálogos orales. No obstante, ninguno de estos lenguajes es necesario para la implementación del otro; por ejemplo, CCXML puede ser integrado con sistemas IVR tradicionales, mientras que VoiceXML puede ser integrado con otros sistemas de control de llamadas.

La adición de funciones avanzadas para el control de llamadas proporcionadas por CCXML supone una nueva gestión y control de los eventos generados por la red telefónica que no son de natural transaccional, pues pueden ocurrir en cualquier momento, independientemente del estado del intérprete de VoiceXML. En esencia, CCXML permite desplazar toda la funcionalidad relacionada con el control de dichos eventos fuera de VoiceXML. De esta forma, VoiceXML se encarga de la interacción con los usuarios mediante diálogos (de naturaleza transaccional) mientras que CCXML se encarga de la interacción con la línea telefónica en sí.

4 CallXML

Similar a CCXML, CallXML es un lenguaje basado en etiquetas creado por Voxeo Community[4], que permite implementar aplicaciones telefónicas basadas en voz convencional o voz sobre IP (VoIP) que interactúan con servidores CallXML, los cuales usan el código de la aplicación para controlar las llamadas telefónicas. Mientras que VoiceXML ha sido diseñado principalmente para proporcionar acceso oral a las aplicaciones web, CallXML ha sido diseñado para facilitar la implementación de aplicaciones web que puedan interactuar con cualquier tipo de llamada telefónica, soportando la iniciación de llamadas salientes, conferencias, interacción entre múltiples llamadas, funciones encuéntreme-sígame, etc.

El lenguaje CallXML incluye etiquetas para realizar acciones en el medio de comunicación, como por ejemplo <playaudio> y <recordaudio> para reproducir/grabar un mensaje durante una llamada. También dispone de etiquetas para realizar acciones con las llamadas, como por ejemplo <answer>, <call> y <hangup>. Asimismo, proporciona etiquetas para realizar operaciones lógicas, como por ejemplo <assign>, <clear> y <goto>, que permiten describir cómo modificar las variables e interactuar con otros lenguajes comúnmente utilizados en los servidores de Internet, como Perl, PHP, ASP, etc.

Por otra parte, CallXML también proporciona etiquetas para gestionar eventos, como por ejemplo <ontermdigit> o <onhangup>, que permiten a la aplicación reaccionar ante acciones que el usuario puede realizar durante una llamada telefónica, como pulsar botones o colgar el teléfono. Asimismo, el lenguaje proporciona etiquetas de bloque para agrupar acciones y eventos de forma lógica.

A modo de ejemplo, la Figura 4 muestra un fragmento de una aplicación "contestador automático" que permite grabar un mensaje y enviarlo por e-mail. Inicialmente, se usa la etiqueta <playaudio> para reproducir el mensaje almacenado en el fichero "saludo.wav", y la etiqueta

[4] http://community.voxeo.com/

<recordaudio> para almacenar el mensaje que se desea enviar. El mensaje se envía a la dirección de e-mail especificada cuando se pulse el botón "#" en el teléfono, el mensaje exceda de 60 segundos de duración, o el usuario permanezca más de 10 segundos en silencio. La etiqueta <text> se usa para generar un mensaje de forma oral mediante la técnica de conversión texto-habla (TTS, text-to-speech) (Rutten y Fackrell, 2003).

```
<?xml version="1.0" encoding="UTF-8" ?>
<callxml>
  <block>
    <playaudio
    value="http://www.servidor.es/saludo.wav"/>

    <recordaudio format="audio/wav"
     value="mailto:alguien@servidor.es"
     termdigits="#" maxtime="60s"
     maxsilence="10s"  beep="TRUE"/>

    <ontermdigit value="#">
     <text>
       Mensaje enviado.
     </text>
    </ontermdigit>

    <onmaxtime>
     <text>
       Se ha excedido el tiempo máximo.
       Mensaje enviado.
     </text>
    </onmaxtime>

    <onmaxsilence>
     <text>
       Se ha permanecido demasiado tiempo en
       silencio. Mensaje enviado.
     </text>
    </onmaxsilence>

  </block>
</callxml>
```

Figura 4. Ejemplo de código CallXML

4.1 Diferencias respecto a VoiceXML

VoiceXML es un lenguaje diseñado para facilitar la labor de los desarrolladores de aplicaciones web que deseen implementar interfaces basadas en RAH, diálogos y TTS para el acceso sus aplicaciones. En cambio, CallXML es un lenguaje diseñado para facilitar la implementación de aplicaciones que necesiten controlar e interactuar con diversos tipos de llamadas basadas en voz (convencional o voz sobre IP), con objeto de controlar el inicio y el enrutamiento de las llamadas, interactuar y responder fácilmente a las entradas efectuadas mediante voz o tonos, y usar otros medios de comunicación, como faxes y vídeo.

Para procesar cada entrada del usuario, VoiceXML utiliza un modelo relativamente complejo basado en formularios (*forms*), campos (*fields*), gramáticas y otros elementos; en cambio, CallXML utiliza un modelo basado en bloques, acciones y eventos, más simple y fácil de aprender, y representable gráficamente mediante un diagrama de flujo.

5 SALT

Al contrario de lo que ocurre con los sistemas de diálogo de interacción exclusivamente oral, los sistemas de diálogo multimodal permiten utilizar varios dispositivos de entrada (p. e. teclado, ratón, micrófono, cámara de visión artificial, pantalla sensible al tacto, etc.), procesándose la información de entrada a varios niveles de abstracción para obtener diversos niveles de comprensión de la misma. Asimismo, estos sistemas permiten utilizar diversos canales de salida para proporcionar información al usuario (p. e. voz, texto, gráficos, etc.) a fin de estimular varios de sus sentidos de forma simultánea (López-Cózar, 2003). Algunos sistemas multimodales permiten incluso que los usuarios puedan elegir entre las diversas modalidades de interacción, permitiendo así una cierta adaptación a las condiciones ambientales de luz, ruido, etc. Además, esta ventaja permite que personas con determinadas discapacidades puedan usar este tipo de sistemas utilizando alguna de las modalidades de interacción disponibles (Beskow et al. 1997).

El lenguaje VoiceXML no permite implementar este tipo de sistemas, pues se basa exclusivamente en el teléfono como canal de comunicación con los usuarios, quienes pueden realizar la entrada oralmente (o mediante tonos) y recibir la salida exclusivamente de forma oral. Además, no permite implementar la sincronización de varios canales de entrada y salida, definiendo las salidas mediante prompts orales en lugar de frames semánticos que puedan ser transformados de forma multimodal.

Para superar estas limitaciones de VoiceXML, Microsoft y otras cinco compañías (entre las que destacan Compaq y SpeechWorks) crearon a finales del año 2001 el foro SALT[5] (*Speech Application Language Tags*) con objeto de estandarizar un lenguaje basado en etiquetas que permita implementar aplicaciones multimodales. La primera versión pública del lenguaje especifica cómo implementar este tipo de aplicaciones usando diversos dispositivos, como navegadores web, PDAs y teléfonos.

Para facilitar la interacción multimodal, SALT permite la entrada del usuario mediante voz o datos, reutilizando en la medida de lo posible los otros lenguajes ya creados por el W3C. De hecho, SALT utiliza la misma especificación de gramáticas para el RAH y el mismo lenguaje para la síntesis de habla (SSML, *Speech Synthesis Markup Language*). Aunque VoiceXML y SALT usan etiquetas distintas, sus núcleos se derivan de otros lenguajes del W3C (como HTML, XML y XHTML), lo que facilita que los desarrolladores familiarizados con el lenguaje HTML y otras tecnologías web pueden usar ambos lenguajes.

La entrada/salida oral se controla mediante el código creado por el programador, cuyas características de bajo nivel proporcionan un gran control y flexibilidad. El lenguaje permite asimismo la entrada mediante DTMF y proporciona diversas funciones para el control de las llamadas telefónicas.

Las etiquetas del lenguaje SALT son objetos caracterizados por una interfaz de propiedades, eventos y métodos, que pueden ser manipulados convenientemente dentro de la aplicación. Su activación generalmente se lleva a cabo mediante un proceso de *disparo*, familiar para muchos desarrolladores de aplicaciones web. Por ejemplo, la etiqueta <prompt> proporciona una interfaz que incluye las siguientes características:

id	propiedad para identificar el objeto.
Start()	método para comenzar reproducción.
oncomplete	evento generado al concluir la reproducción.

La etiqueta <listen> constituye el bloque básico para realizar el RAH. También consta de un identificador (id) y un método Start(), junto con las siguientes características:

<grammar>	gramática para reconocer la entrada oral.
<bind>	directiva para asignar la respuesta del usuario a una variable del documento.
onreco	evento generado cuando el RAH se realiza con éxito.

A modo ilustrativo, la Figura 5 muestra un fragmento de código escrito en lenguaje SALT, que genera un mensaje de bienvenida, solicita una clave (PIN) al usuario (mediante el prompt "askPIN"), y activa un elemento <listen> llamado "recoPIN" para realizar el proceso de RAH usando la gramática "PINdigits.grxml". Si dicho proceso se realiza con éxito, el método bind asigna la respuesta del usuario a la variable "iptPIN", y el manejador de eventos onreco envía la clave (PIN) proporcionada por el usuario a un servidor web.

```
<html xmlns:salt="http://www.servidor.es/SALT">
 <body onload="sayWelcome.Start()">

  <form id="PIN" action="checkPIN.html">
            <input id="iptPIN" type="text" />
  </form>

  <salt:prompt id="sayWelcome" oncomplete=
  "askPIN.Start(); recoPIN.Start()">
    Bienvenido a mi aplicación.
  </salt:prompt>

  <salt:prompt id=""askPIN">
    Por favor,  diga su clave.
  </salt:prompt>

  <salt:listen id="recoPIN" onreco="PIN.submit()">
     <salt:grammar src="PINdigits.grxml" />
     <salt:bind targetElement="iptPIN" />
  </salt:listen>

 </body>
</html>
```

Figura 5. Ejemplo de código SALT

Dado que SALT permite utilizar otros dispositivos de comunicación adicionales al teléfono, como por ejemplo PDAs, el proceso de RAH se puede iniciar automáticamente cuando se producen

[5] http://www.saltforum.org

diversos eventos multimodales, por ejemplo, cuando el usuario hace clic con el puntero (*stylus*) del PDA en un campo de texto, activándose una gramática de reconocimiento adecuada para el dato correspondiente a ese campo.

5.1 Diferencias respecto a VoiceXML

Como se ha comentado previamente, VoiceXML permite la entrada el usuario mediante habla y DTMF, mientras que SALT permite la entrada mediante habla, DTMF y datos. Con respecto a los dispositivos, VoiceXML sólo permite usar el teléfono, mientras que SALT permite usar el teléfono, navegadores web, PDAs y teléfonos inalámbricos. Por otra parte, VoiceXML sólo permite la interacción mediante el único modo permitido por el teléfono, en cambio, SALT permite la interacción multimodal simultánea mediante voz, DTMF y datos, gracias a la disponibilidad de un mayor rango de dispositivos utilizables.

6 Conclusiones

VoiceXML es un lenguaje basado en etiquetas que hace tan fácil crear aplicaciones basadas en voz como lo es crear páginas web. El consorcio W3C aceptó VoiceXML como el estándar basado en XML para crear interfaces basadas en voz y DTMF. El estándar ha sido desarrollado para operar sobre la infraestructura web existente, reduciendo las barreras existentes en el desarrollo de aplicaciones basadas en voz mediante una alternativa a las tecnologías IVR (*Interactive Voice Response*) tradicionales.

No obstante, a pesar de sus numerosas ventajas, el estándar también presenta numerosas limitaciones para aplicaciones comerciales. Por ello se han desarrollado con posterioridad otros lenguajes como CCXML, CallXML y SALT destinados a completar las carencias del estándar en cuanto a control de las llamadas telefónicas e interacción multimodal.

La aparición de estos lenguajes proporciona ventajas para proveedores de servicios y clientes. Las aplicaciones desarrolladas gracias a ellos pueden ser usadas en un gran número de servicios comerciales, como por ejemplo, finanzas, seguros, viajes, entretenimiento y servicios de directorio. Estas aplicaciones incrementan la satisfacción de los clientes, proporcionando un acceso rápido a la información y ofreciendo un servicio personalizado que, en algunos casos, puede llegar a ser similar al proporcionado por un operador humano, objetivo final de los sistemas de diálogo.

7 Referencias

Abbott, K. R. 2002. Voice enabling web applications. VoiceXML and beyond. a! press

Billi, R., Castagneri, G., Danielli, M. 1997. Field trial evaluations of two different information inquiry systems. Speech Communication, 23, 1-2, pág. 83-93

Beskow J., McGlashan S. 1997. Olga - A Conversational Agent with Gestures. Proceedings the IJCAI'97 workshop on Animated Interface Agents - Making them Intelligent, Nagoya, Japan

Campillo, F., Banga, E. R. 2003. On the design of cost functions for unit-selection speech synthesis. *Proc.* Eurospeech, pág. 289-292

Hain T., Woodland P.C., Niesler T.R., Whittaker E.W.D. 1999. The 1998 HTK System for Transcription of Conversational Telephone Speech; Proc. International Conference on Acoustics, Speech and Signal Processing

López-Cózar, R., García, P., Díaz, J., Rubio, A. J. 1997. A voice activated dialog system for fast-food restaurant applications. *Proc.* Eurospeech, pág. 1783-1786

López-Cózar R. 2003. Uso de canales adicionales en los sistemas conversacionales. Procesamiento del Lenguaje Natural, nº 30, pág. 89-97

Rabiner L. R., Juang B. H. 1993. Fundamentals of Speech Recognition. Prentice-Hall

Rutten, P., Fackrell, J. 2003. The application of interactive speech unit selection in TTS systems. *Proc.* Eurospeech, pág. 285-288

Seneff, S., Polifroni, J. 2000. Dialogue management in the Mercury flight reservation system. *Proc.* ANLP-NAACL 2000 Satellite Workshop, pág. 1-6

Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T., Hetherington, L. 2000. Jupiter: A telephone-based conversational interface for weather information. IEEE Trans. on Speech and Audio Proc., 8(1), pág. 85-96